

✓
IEN: 55
Section: 2.4.2.1

September 1978

RECEIVED
SEP 1978
J. F. SHOCH

SPECIFICATION OF INTERNETWORK TRANSMISSION CONTROL PROTOCOL

TCP
Version 4

Jonathan B. Postel

September 1978

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, California 90291

(213) 822-1511

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA



September 1978

IEN:55

Section: 2.4.2.1

Replaces: IENs 44, 40, 27, 21, 5

Transmission Control Protocol

Version 4

1. INTRODUCTION

The Transmission Control Protocol (TCP) is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and especially in interconnected systems of such networks.

This document describes the functions to be performed by the Internetwork Transmission Control Protocol (TCP), the program that implements it, and its interface to programs or users that require its services.

1.1. History

There have been four previous TCP specifications: The first [CDS74] defined version 1 of TCP. A second [PGR76a] was written for the Defense Communications Agency in connection with its AUTODIN II project. The third [Cerf77] defined version 2, for use in the ARPA internetwork research projects. The fourth [CP78] defined version 3, a refinement of version 2.

The AUTODIN II version differed from the original version in the following ways:

Specification of a resynchronization mechanism was included, and fields for security and priority, which were known requirements of AUTODIN II, were added.

The first Internet version (version 2) differed from the original version in the following ways:

A different resynchronization procedure was introduced; an "option" field was defined for the TCP header to accommodate not only security and priority but other special features concerned with, for example, segment speech services, diagnostic timestamping, and so on.

Version 2 eliminated all error messages but for RESET, and thus simplified the header format. There are still many local errors which can be reported to the user, but none of these need cross the network(s) between TCP's.

Connection closing was slightly more elaborate in Version 2 than in version 1 because the

TCP-4

Introduction

FIN signals had to be acknowledged. Furthermore, the INT and FIN facilities no longer caused flushing of the data stream. (A separate "flush" facility was tested, but eliminated in the end.) Dealing with flow-control windows that have gone to zero was a new feature of version 2; and, finally, the reassembly of fragments into segments was more carefully specified.

In version 3 TCP further evolved. The primary changes from version 2 were as follows:

The resynchronization mechanism was eliminated in favor of a quiet period on initialization of the TCP.

Buffer management and letters boundaries were more tightly coupled associated by the coupling of the end of letter flag to a receive buffer size.

The interrupt signal was eliminated in favor of an urgent pointer.

A further separation of the internet and TCP specific information in the segment format was achieved.

Version 4 TCP specified here has the following changes:

* [The TCP now expects to call on a lower level protocol module in the host for certain functions; in the general Internetwork case, TCP expects the lower level module to implement the ARPA Internetwork Protocol [Postel78d] or something functionally equivalent to it.

Addressing information necessary to reach a specific TCP implementation is expected to be carried on the lower level protocol.

Fragmentation and reassembly have been eliminated from TCP and made the responsibility of the lower level protocol.

1.2. Scope

The TCP is intended to provide a reliable process-to-process interprocess communication service in a multinet environment. The TCP is intended to be a host-to-host protocol in common use in multiple networks.

1.3. Other Documentation

For other documentation, see the items cited in the History Section (1.1) and the items listed in the Bibliography.

John Davidson reported on BBNs UNIX TCP. It is now running TCP 2.5, and should be running IN-4, and TCP-4 by 1 Nov, really aiming for 1 Oct.

BBN's EON work may use the DTI version of TCP which is in C. This project, led by Wingfield, expects to have a C version for EON by 1 Jan 79. This version will not be the same as the one used for the ARPA Internet project.

ii) MIT - Dave Reed

Dave reported that a Multics implementation of TCP version 3.1 was nearly completed, and now work is in progress on version 4. User side is straightforward, but there are policy problems to install server. MIT is also trying to get a UNIX TCP, an ITS TCP, and a TOPS 2040 TCP. On the LCS NET progress is being made. The 3rd interface has been ordered, and testing is now underway between two machines. A key problem at MIT is a shortage of IMP ports; in fact, they currently have two more hosts than ports.

iii) XEROX-PARC - John Shoch

John reported on PARCs experiment using the PRNET as transit net between two ETHERNETs. PARC now has about 22-25 net's (lost track of numbers). PARC is also doing a packet speech experiment using BYTE STREAM connection - up to 500 KB - so unencoded speech is sent.

iv) SRI - Jim Mathis

Jim reported that he has not really done much about converting to the new version of TCP and IN. He is waiting to see if version 4 turns into version 4.1. On the Port Expander idea things are progressing slowly also, the current thing works with the 32 bit 1822 leaders.

v) UCL - Andrew Hinchley

Andrew reported that the FTP standard from the EPSS group will be brought up on a Tenex so that experiments with end-to-end FTP between EPSS hosts and an ARPANET host can be performed, and possible experiments with hosts in other in X.25 networks.

Postel

[page 6]

21 August 1978

IEN: 53

Internet Meeting Notes

vi) NDRE - Yngvar Lund

Yngvar discussed the state of development at NDRE. A TCP-3 is near completion for the NORD-10 and is nearly ready for testing. The need for a TCP on ELF was pointed out.

vii) CCA - Kou Mei Chuang

Kou Mei reported on the progress at CCA. Currently they are converting a TCP-11 supplied by Jim Mathis to run under RSX-11. This is version 2.5. When Jim can supply a version 4, they will cover that.

It is clear that Jim Mathis is the leader on TCP-4 & IN-4 for pdp-11s. Everyone that is waiting for his program to be available should contact him.

viii) LL - Jim Forgie

Jim Forgie remarked that speech conferencing had been demonstrated in SATNET with SIMP-1. SIMP-3 is up now, and speech is unusable due to now delay problems. There is a need to do internet speech

8/78

Lincoln Labs

-- Messages from file: [PARC-MAXC]<SHOCH>MESSAGE.TXT;1
-- TUESDAY, OCTOBER 10, 1978 20:53:27-PDT --

Date: 9 OCT 1978 1951-PDT
From: POSTEL at USC-ISIB
Subject: TCP Meeting Notes
To: [ISIE]<Postel>TCP-INTERNET.List:

The following 26 pages are the meeting notes from the TCP meeting held at SRI 18 & 19 September. The notes are so long because they include as a set of appendices the memos detailing the TCP implementation status distributed at the meeting. The notes are also in the file
--
<INTERNET-NOTEBOOK>TCP-MEETING-NOTES.TXT at ISIE.

--jon.

Jon Postel
ISI
9 October 1978

TCP Meeting Notes - 18 & 19 September 1978

Editors Note

This meeting was billed as a TCP testing session. The first hour or so of the meeting was taken up with a discussion of what the various implementors present expected their programs to be able to accomplish with the programs of the other implementors.

X { It became apparent that almost no useful demonstrations could be performed due to the differences between the implementations.

The planned agenda was discarded, and the meeting turned primarily into a discussion of the specifications of TCP with a few remarks about testing procedures.

Much of the detail of the discussion of the TCP specifications will be omitted from these notes, but the information will be conveyed to the specification editor for use in the next edition.

Discussion of the State of TCP Implementations

The TCP implementations available for testing at the meeting were:

SRI:	TCP 2.5
BBN Tenex/Tops 20:	TCP 2.X
UCLA:	TCP 4
BBN Unix:	TCP 2.5
MIT Multics:	not ready

The state of BBN's Tenex and Tops 20 versions was explored in some detail:

Version	Comments
2.5-e	Tested; BCPL, running at ISI; "old faithful"; has resynchronization ARQ, RSN, INT, etc.
2.5	Tested; BCPL & Monitor versions; running at SRI-KA; INT, RSN, ARQ removed, code cleaned up.
2.5+e	Tested; Hand coded; running at BBND, BBNC; bugs fixed.

Postel

[page 1]

9 October 1978

TCP Meeting Notes

2.5.2c

Untested; fixes installed to attempt cures for all

This work will commence when SRI TCP-4 is released and will be coterminous with the CCA revision of this software.

✓ PARC - Shoch

John briefly reviewed the status of networks at XEROX. There are 7 types of nets, 20 nets, 500 hosts. The particular experiment of interest to the IN group is the use of the PRNET as a link between two ETHERNETs located about a mile apart. Adding the Radionet to the existing Parc internetwork architecture required building an 1822 interface, and adding a network-specific driver for the Radionet in the gateway at each end. This driver performs network specific fragmentation, splitting up large internet packets into smaller fragments that can be sent through the Radionet.

Over 1000 hours of gateway operation have been accumulated, and the system seems to work well; the radio link can support one-way byte stream traffic at about 12 KB/S. John noted that measurement shows that they can get about 0.3 KB/S from a line rated at 9.6 KB/S; and out of the 100 KB/S radio channel, they can usually get 12.5 KB/S; and if all the parameters are tuned, they can on occasion get 25 to 30 KB/S.

John also discussed a flow control problem that can seriously effect throughput. Basically, the variability in delay may trigger undesirable retransmissions that result in every message being sent twice. These effects may be repeated resulting in every message being sent many times. The serious part of the problem is that such states may be stable. This topic is further discussed later in the meeting.

This particular failure mode has since been corrected, by slightly modifying the flow control heuristics, and system behaves properly. This experience again highlights the difficulty of designing good flow control and congestion control mechanisms, especially when inter-connecting networks whose basic performance differs by several orders of magnitudes.

John also requested contributions to a bibliography on local networks that he is preparing.

1/25/79

Internet Meeting

@ ISI

Postel

[Page 6]

IEN 76

Internet Meeting Notes

MITRE - Skelton

Anita reported that the MITRE-Unix is now up, and the local network is installed with 6 interface units in place. The local net is based on cable TV technology using a 300 MB base band with a 300 KB subchannel for packet data transmission. TTY-to-TTY connections have been tested on the cable, and the 11/70 is being connected via a single DH-11 line with software multiplexing. Eventually, the 11/70 will interface to the cable via a UMC-Z80. There are three LSI-11s soon to be connected to the cable. These could run the MOS-TCP from Jim Mathis. Anita will provide some material from a MITRE working paper for an IEN describing the network.

FORD - Ken Biba

Ken discussed the work at FORD on the KSOS (Kernelized Secure Operating System). This is to be a UNIX-like system with multi-level security. The Critical Design Review is at the end of February. The system will be written in MODULA (possible fall back is "C"). Ken pointed out the problem of handling IN protocol fragment reassembly in a secure system. This is further discussed in IEN-73. The key issue seems to be to provide some demultiplexing information in every fragment so as to not require "shared" storage in the kernel.

Ken also reported that FORD is building a local net based on an 1MB channel with DMA connections to PDP-11s. The network hosts will use NCPs. The network interface is based on a UMC-Z80 from ACC.

FINAL REPORT OF THE STANFORD

UNIVERSITY TCP PROJECT

Vinton G. Cerf
1 April 1980

ABSTRACT:

This report provides a brief historical and technical summary of the Stanford University internet, host-to-host Transmission Control Protocol (TCP) project. Developed from 1973-1976 at Stanford, Bolt Beranek and Newman and University College London, the effort then continued at other research and development centers during 1977-1980. Originally designed as a monolithic internet protocol, the internet aspects were separated into a distinct protocol layer in early 1979 with the publication of version 4 of TCP. The resulting pair of protocols, TCP4 and Internet Protocol (IP) are now undergoing procedures for standardization within the DoD and Intelligence Community. The four most valuable results of the Stanford effort were the assessment of which functions could or should be implemented in the protocol and which should not, the implementation of an efficient, assembly-language version of TCP for an LSI-11 computer, the development of a "micro" operating system (MOS) for the same computer, and the specification of the TCP protocol.

1. Introduction:

The Stanford TCP project was supported by the Defense Advanced Research Projects Agency (DARPA) under contract No. DAHC 15-73-C-0363 and MDA903-76-C-0206, ARPA Order No. 2494 during the period 1 July 1973 to 30 September 1976. During the time that the project was at Stanford, two versions of TCP were developed, one in December 1974 [1] and another which was published in March 1977 [2], by DARPA. Since that time, two other versions have been published, versions 3 and 4 [3,4] in January 1978 and February 1979, respectively. Editing of these latter versions was the responsibility of J. Postel, Information Sciences Institute, University of Southern California. Intermediate versions were published in July 1976 [5], July 1978 [6], and August 1979 [7]. The July 76 version was developed for the Defense Communication Agency by J. Postel, L. Garlick and R. Ram of Stanford Research Institute for the Defense Communication Agency in connection with the AUTODIN II project. The final versions of TCP and Internet Protocol were published in January, 1980 [14,15] by DARPA on behalf of the Defense Communication Agency.

During the course of the DARPA Internetworking program, which supported the TCP development, a great many people and groups were involved in or influenced the development of the TCP.

The initial impetus for the effort resulted from work by R. Kahn and V. Cerf, which was published in May 1974 [8], but whose basic roots were already known in September 1973 [9].

An initial design for TCP was worked out during 1973-74 at Stanford among V. Cerf, Y. Dalal, C. Sunshine and R. Karp, with the participation of R. Tomlinson [Bolt Beranek and Newman], W. Plummer [BBN], R. Metcalfe [Xerox PARC] and D. Boggs [Xerox PARC]. Implementation, testing and further development were carried out jointly at Stanford with J. Mathis and J. Estrin, Bolt Beranek and Newman, and University College London [F. Deignan, C. J. Bennett, A. J. Hinchley and M. Galland]. Visiting at Stanford during this initial development period were G. LeLann [University of Rennes, France] and D. Belsnes [University of Oslo] who provided additional philosophical leavening which influenced the design of the protocol.

By 1976, implementations had been written for the Tenex/PDP-10 [Tomlinson, Plummer - in BCPL], ELF/PDP-11 [Tomlinson, Plummer - in BCPL; Karp, Dalal, Cerf - BCPL], LSI-11 [Mathis - assembly language], PDP-9 [Deignan - BABBAGE] and some performance experience obtained [10].

Since then, implementations have been pursued for UNIX [M. Wingfield - "C"; J. Haverty - assembly language], OS 360 [B. Braden - assembly language], Multics [D. Clark - PL/1], TOPS-20 [W. Plummer - assembly language], and NORD-10 [A. Stensby].

2. TECHNICAL ISSUES

The initial concept behind TCP was very simple. Two processes wishing to communicate had only to know the other's "address". Data would be accounted for in 8-bit octets, and sequence numbers would be used to re-order the received data at the destination, if necessary. The first packet would have a special synchronization flag ["SYN"] which would alert the receiver that the sender's sequence numbers would start with the one associated with the "SYN" packet. All control information would be associated with data sequence numbers so that end to end acknowledgements for data could also be used to acknowledge control. If resynchronization were needed, the sender could simply send another "SYN" packet. There would be no need for a "connection set-up" in the conventional sense. Finally, packet formats would permit internet gateways to fragment packets into identically formatted, smaller packets if necessary, to get through a net with a smaller packet size. Reassembly of fragments would be done by the destination.

a hazard if the host failed just as the "next sequence number" to use fell into the so-called initial sequence number "forbidden zone". A complex set of tests were defined to catch this case, but required action on each packet transmitted to determine whether the hazardous "forbidden" zone had been entered. If so, the sequence numbers on the connection would be re-synchronized to skip around the danger area. The probability of the hazard actually occurring was judged to be quite small, and finally these tests were eliminated, vastly simplifying the TCP definition and implementation.

The third critical issue had to do with the treatment of packets which required fragmentation to pass through a network. Gateways between nets were postulated which could detect that an incoming packet was too large to fit encapsulated in the packet format of the next network.

A fragmentation strategy was developed which permitted an internet packet to be fragmented into smaller packets and marked in such a way that the resulting fragments could be routed independently of one another and could still be reassembled at the final destination.

A major change to the TCP philosophy occurred when the basic internetting functions (addressing, fragmentation and type of service selection) were separated from the end to end functions of TCP (sequencing, retransmission, duplicate detection, flow control and port multiplexing). At this point, the fragmentation problem was substantially simplified since it applied only to the internet packets and limited knowledge of protocols to the IP layer in the gateway.

At the same time, this permitted other higher level protocols, adjacent to the TCP layer, to rely on the special services, including fragmentation and reassembly, of the IP protocol layer.

Network Security

The TCP concepts were applied to the ARPA network security program and an architecture was developed which accommodated the use of TCP as an end to end protocol, below which, end to end encryption could take place. This architecture became even simpler when the TCP was split from the internet protocol functions since security was provided at the IP level, allowing many different transport protocols, in addition to TCP, to be secured by the same basic system.

Implementation and Experimentation

Two TCPs were developed during this project:

1. BCPL for PDP-11 under ELF operating system.